

Claims

What is Claimed is:

1. A pre-processor for processing a program that includes at least one function call, comprising:
 - a code expander for detecting the at least one function call of the program;
 - at least one trace point; and
 - an inserter, communicably connected with the code expander and the at least one trace point, for inserting the at least one trace point in the program relative to the at least one function call.
2. The pre-processor of claim 1, wherein the at least one trace point invokes detection of a characteristic regarding the at least one function call, during run-time execution of the program.
3. The pre-processor of claim 1, further comprising:
 - a function routine having a characteristic, operable upon execution of the program and encountering the at least one function call;
 - wherein the relativeness of the insertion of the at least one trace point with respect to the at least one function call is dictated, at least in part, by the characteristic of the function routine.

4. A method of pre-processing a program that includes at least one function call, comprising the steps of:

detecting the at least one function call of the program; and
associating at least one trace point with the at least one function call.

5. The method of claim 4, wherein the step of associating includes:
inserting the at least one trace point in the program relative to the at least one function call.

6. The method of claim 4, wherein the relativeness of insertion of the at least one trace point with respect to the at least one function call is dictated, at least in part, by a characteristic of a function routine corresponding to the at least one function call.

7. A run-time tracer, comprising:
a data processor;
a database communicatively connected to the data processor;
a software program operating on the data processor, having at least one trace point associated with at least one function call of the program; and
a trace id stored in the database, that corresponds to and identifies the at least one trace point, associated with the at least one function call of the program.

8. The run-time tracer of claim 7, wherein the software program is object code readable by the data processor.

9. The run-time tracer of claim 8, further comprising:
a trace information dictated by occurrences at the at least one function call during execution of the program; and
a shared memory for maintaining the trace information.
10. The run-time tracer of claim 9, wherein the database includes at least one trace id and relates the trace id with the at least one trace point.
11. The run-time tracer of claim 10, wherein the database maintains a human-readable information that corresponds to the trace information and the at least one trace id and relates the trace information and the at least one trace id to a portion of the human-readable information indicative of occurrences associated with the trace information during execution.
12. The run-time tracer of claim 10, wherein the shared memory is written to by and during multiple executions of the program.
13. The run-time tracer of claim 11, further comprising:
a snapper for instantaneously capturing data of at least a portion of the shared memory.

14. The run-time tracer of claim 13, wherein the associatedness of the trace id and the at least one function call of the program is achieved by the database relating the trace id to the trace point corresponding to the at least one function call, as captured by the snapper.

15. The run-time tracer of claim 7, further comprising:

a function routine accessible to the data processor, corresponding to and called by the at least one function call;

a characteristic of the function routine, wherein the characteristic is detectable via the data processor on execution of the function routine;

a trace channel accessible to the data processor; and

an identifier of the characteristic that is saved to the trace channel via operation of the data processor, upon detection of the characteristic.

16. The run-time tracer of claim 15, wherein the trace channel is a circular buffer written to by copy direct from the data processor.

17. A shared memory for run-time tracing, comprising:

an executable program, including a function call to a function routine and a trace point corresponding to and identifying the function call; and

an address of the shared memory wherein the executable program writes trace information associated with the trace point during program run-time.

wherein the shared memory is simultaneously accessible at the address to multiple instances of the executable program during multiple runs.

18. The shared memory of claim 17, wherein the shared memory is selectively configured during execution of the executable program to capture select ones of the trace points, in order to permit tracing via the run-time tracing of only select ones of the trace points.

19. The shared memory of claim 18, further comprising a control channel of the shared memory, wherein a data in the control channel dictates the select ones of the trace points involved in the run-time tracing.

20. The shared memory of claim 17, wherein the shared memory is concurrently connected and accessible to a plurality of run-time instances of execution of the program.

21. The shared memory of claim 20, wherein the shared memory contains trace information related to trace points encountered in a most current run-time instance of execution of the program, as well as past instances.

22. A method of run-time tracing, comprising the steps of:
processing a data;
storing the data in a memory;

programming the step of processing to perform a function routine, so that at least one trace point is associated with at least one function call of the function routine; and

storing a trace id in a database, the trace id serving to uniquely identify the at least one trace point.

23. The method of claim 22, wherein the step of programming is performed via an object code readable in the step of processing, and wherein the object code includes the at least one trace point.

24. The method of claim 23, further comprising the step of:
relating the trace id to the at least one trace point via the database.

25. The method of claim 25, wherein the memory is communicatively accessible and shared by a plurality of independent steps of processing.

26. The method of claim 22, further comprising the step of reporting a detected characteristic encountered in performance of the function routine, via identification by relation to the trace id.

27. A method of sharing memory for run-time tracing, comprising the steps of:
executing a program, the program includes a function call to a function routine and a trace point corresponding to and identifying the function call;

writing to a memory communicatively connected to other steps of
executing; and
capturing a state of the memory.

28. The method of claim 28, further comprising the step of selectively
accessing the shared memory during the step of executing, in order to trace select ones of
a plurality of unique ones of the trace points.

29. The method of claim 29, further comprising the step of controlling the
program during the step of executing in order to effect the step of selectively accessing.

30. The method of claim 30, further comprising the steps of concurrently
connecting to and accessing the shared memory by independent and plural steps of
executing the program.